

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **11096054 A**(43) Date of publication of application: **09.04.99**

(51) Int. Cl.

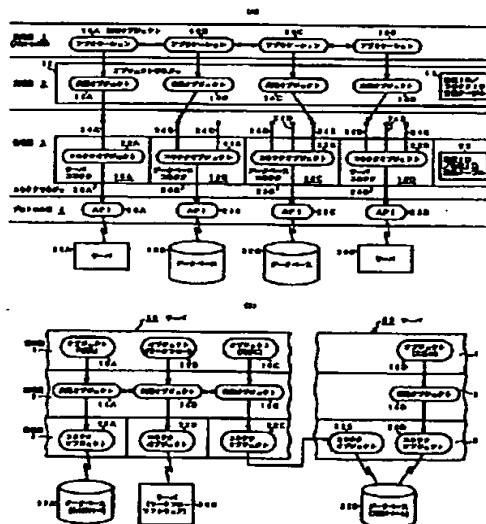
**G06F 12/00**  
**G06F 17/30**(21) Application number: **09253083**(22) Date of filing: **18.09.97**(71) Applicant: **FUJITSU LTD**(72) Inventor: **MATSUZUKA TAKAHIDE**  
**HARA HIROTAKA**  
**KANETANI NOBUYUKI****(54) SYSTEM FOR CONSTRUCTING DATA BASE**  
**INTEGRATING APPLICATION**

COPYRIGHT: (C)1999,JPO

(57) Abstract:

**PROBLEM TO BE SOLVED:** To attain a complete distributed environment capable of equivalently handling storing positions in an object data base or a server without being conscious of them.

**SOLUTION:** An operation application is constructed by loading plural logical objects 10A to 10D on a logical layer 1, common objects 14A to 14D corresponding to the logical objects 10A to 10D are formed on a realization layer 2 and acquires connector ID by referring to a 1st conversion table 16 based on logical ID. The common objects 14A to 14D operate data by using service provided by interfaces 24A to 24E on a physical layer 3. The physical layer 3 acquires physical ID by referring to a 2nd conversion table 25 based on the logical ID, reads out a real object from a data base, generates connector objects 22A to 22D and links the objects 22A to 22D to the logical objects 10A to 10D through the common objects 14A to 14D.



(19) 日本国特許庁 ( J P )

(12) 公開特許公報 ( A )

(11) 特許出願公開番号

特開平 1 1 - 9 6 0 5 4

(43) 公開日 平成 1 1 年 ( 1 9 9 9 ) 4 月 9 日

(51) Int. Cl.

G06F 12/00

17/30

識別記号

512

庁内整理番号

F I

G06F 12/00

15/40

512

380

E

技術表示箇所

審査請求 未請求 請求項の数 1 5 O L ( 全 2 1 頁 )

(21) 出願番号 特願平 9 - 2 5 3 0 8 3

(22) 出願日 平成 9 年 ( 1 9 9 7 ) 9 月 1 8 日

(71) 出願人 0 0 0 0 0 5 2 2 3

富士通株式会社

神奈川県川崎市中原区上小田中 4 丁目 1 番  
1 号

(72) 発明者 松塚 貴英

神奈川県川崎市中原区上小田中 4 丁目 1 番  
1 号 富士通株式会社内

(72) 発明者 原 裕貴

神奈川県川崎市中原区上小田中 4 丁目 1 番  
1 号 富士通株式会社内

(72) 発明者 金谷 延幸

神奈川県川崎市中原区上小田中 4 丁目 1 番  
1 号 富士通株式会社内

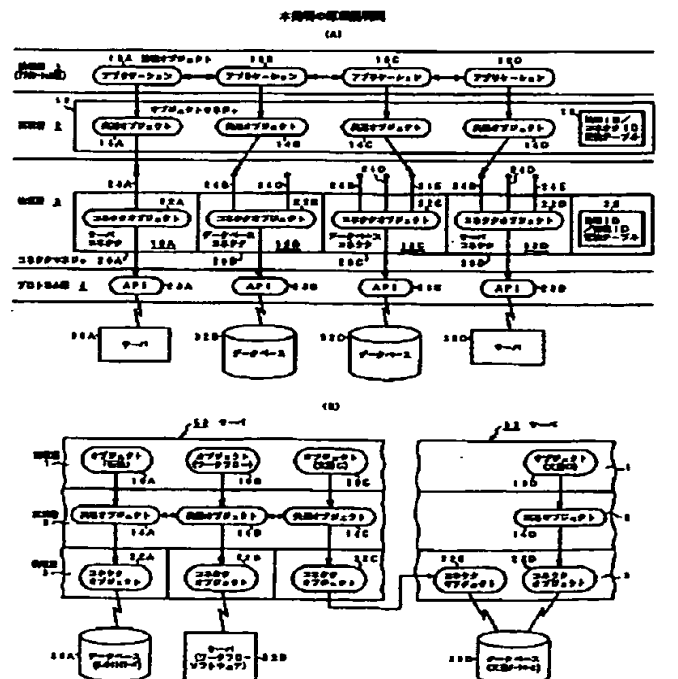
(74) 代理人 弁理士 竹内 進 ( 外 1 名 )

(54) 【発明の名称】 データベース統合アプリケーション構築システム

(57) 【要約】

【課題】 オブジェクトのデータベースやサーバでの格納位置を意識せずに等価的に扱うことができる完全な分散環境を実現する。

【解決手段】 論理オブジェクト 10A-10D を論理層 1 に実装して業務アプリケーションを構築し、実現層 2 は論理オブジェクト 10A-10D に対応して共通オブジェクト 14A-14D を生成し、論理 ID により第 1 変換テーブル 16 を参照してコネクタ ID を取得する。共通オブジェクト 14A-14D は、物理層 1 がインタフェース 24A-24E で提供するサービスを使ってデータを操作する。物理層 3 は論理 ID により第 2 変換テーブル 25 を参照して物理 ID を取得し、データベースの実オブジェクトを読取ってコネクタオブジェクト 22A-22D を生成し、これを共通オブジェクト 14A-14D を経由して論理オブジェクト 10A-10D にリンクさせる。



## 【特許請求の範囲】

【請求項 1】操作方法の異なる複数のデータベースに存在するデータをオブジェクトとして扱い、且つオブジェクト格納位置を意識せずに同じ操作方法で等価的に扱うためのデータベース統合アプリケーション構築システムに於いて、

基本アーキテクチャとして論理層、実現層、物理層、及びプロトコル層を構築し、

前記論理層に、組織、文書等の複数種類の論理オブジェクトのロジックを実装してアプリケーションを構築し、  
前記実現層に、前記論理オブジェクトに対応して共通オブジェクトを生成し、前記論理オブジェクトの一意性を示す論理 ID により第 1 変換テーブルを参照してコネクタ ID を取得し、更に前記共通オブジェクトに前記物理層の提供するインタフェースを使用してデータを操作させるオブジェクトマネージャを設け、

前記物理層に、前記コネクタ ID で選択され、前記論理 ID により第 2 変換テーブルを参照して得た物理 ID により前記データベースから実オブジェクトを取得してコネクタオブジェクトを生成し、該コネクタオブジェクトを予め準備したインタフェースにより前記実現層の共通オブジェクトを経由して前記論理層の論理オブジェクトにリンクさせるコネクタを設け、

前記プロトコル層に、前記コネクタが操作するデータベース又はサーバとの通信を確立するアプリケーション・プログラム・インタフェースを設けたことを特徴とするデータベース統合アプリケーション構築システム。

【請求項 2】請求項 1 記載のデータベース統合アプリケーション構築システムに於いて、前記コネクタは、処理要求を受け取ったときに起動する処理手続となる特定の機能をもったメソッドの集合でなる 1 又は複数のインタフェースを実装し、実装されているインタフェースを前記実現層に公開して使用させることを特徴とするデータベース統合アプリケーション構築システム。

【請求項 3】請求項 1 記載のデータベース統合アプリケーション構築システムに於いて、前記物理層のコネクタとしてデータベースコネクタとサーバコネクタを実装し、

前記データベースコネクタは、前記第 2 変換テーブルから得られた物理 ID で指定されるデータベースにアクセスして前記論理 ID で特定される実オブジェクトを取得してコネクタオブジェクトを生成し、

前記サーバコネクタは、前記第 2 変換テーブルから得られた物理 ID で指定されるサーバにアクセスして前記論理オブジェクトの論理 ID で特定される実オブジェクトを前記サーバの管理するデータベースから取得してコネクタオブジェクトを生成することを特徴とするデータベース統合アプリケーション構築システム。

【請求項 4】請求項 3 記載のデータベース統合アプリケーション構築システムに於いて、前記サーバコネクタ

は、前記物理 ID で特定されたサーバに自己のデータベースで使用しているインタフェースを問い合わせ、前記実現層の共通オブジェクトに提示することを特徴とするデータベース統合アプリケーション構築システム。

【請求項 5】請求項 1 記載のデータベース統合アプリケーション構築システムにおいて、前記コネクタは、データにアクセスするための基本インタフェースを実装すると共に、必要に応じて永続性機能、バージョン管理機能、ファイル操作機能、アクセス制御機能等のインタフェースを予め定義して前記実現層の共通オブジェクトにより操作させることを特徴とするデータベース統合アプリケーション構築システム。

【請求項 6】請求項 1 記載のデータベース統合アプリケーション構築システムに於いて、前記データベース又はサーバのデータを前記コネクタによりキャッシュしてデータのアクセス時間を低減することを特徴とするデータベース統合アプリケーション構築システム。

【請求項 7】請求項 1 記載のデータベース統合アプリケーション構築システムに於いて、前記コネクタでデータをオブジェクトに変換することにより、オブジェクト指向に対応していないデータベースをオブジェクト指向データベースとして使用することを特徴とするデータベース統合アプリケーション構築システム。

【請求項 8】請求項 1 記載のデータベース統合アプリケーション構築システムに於いて、前記実現層のオブジェクトマネージャが共通オブジェクトを移動することにより、同じコンピュータまたは異なるコンピュータのデータベース間のオブジェクトの移動を実現することを特徴とするデータベース統合アプリケーション構築システム。

【請求項 9】請求項 1 記載のデータベース統合アプリケーション構築システムに於いて、前記実現層のオブジェクトマネージャが共通オブジェクトを複製することにより、同じコンピュータまたは異なるコンピュータのデータベース間のオブジェクトの複製を実現することを特徴とするデータベース統合アプリケーション構築システム。

【請求項 10】請求項 1 記載のデータベース統合アプリケーション構築システムに於いて、前記実現層のオブジェクトマネージャが前記物理層のコネクタを動的に追加又は削除することにより、接続するデータベースやサーバを動作中に変更可能とすることを特徴とするデータベース統合アプリケーション構築システム。

【請求項 11】請求項 1 記載のデータベース統合アプリケーション構築システムに於いて、前記コネクタに新機能を追加する場合、該新機能をサポートするインタフェースを設計して前記コネクタに実装することを特徴とするデータベース統合アプリケーション構築システム。

【請求項 12】請求項 11 記載のデータベース統合アプリケーション構築システムに於いて、前記コネクタで新

3

機能としてサポートしたインタフェースを前記実現層で操作できない場合、実現層は前記新機能のインタフェースを前記論理層に公開し、前記論理層から実現層を介さずに直接前記コネクタのインタフェースを操作することを特徴とするデータベース統合アプリケーション構築システム。

【請求項 1 3】請求項 1 記載のデータベース統合アプリケーション構築システムに於いて、前記論理層のオブジェクトにより特定データを操作することにより、他のコンピュータのサービスを起動し、該コンピュータ間で非同期にメッセージをやりとりしてサービス依頼側の待ち時間を低減させることを特徴とするデータベース統合アプリケーション構築システム。

【請求項 1 4】請求項 1 3 記載のデータベース統合アプリケーション構築システムに於いて、前記実現層にイベントオブジェクトを実装し、該イベントオブジェクトのプロパティに所定の値を代入することにより、イベントが発生したことを他のコンピュータのイベントマネージャに伝えて対応するルールを起動させることを特徴とするデータベース統合アプリケーション構築システム。

【請求項 1 5】請求項 1 記載のデータベース統合アプリケーション構築システムに於いて、前記論理層に複数の論理オブジェクトを実装することにより、複数のデータベースやコンピュータにまたがって存在するデータを一度に検索・更新することを特徴とするデータベース統合アプリケーション構築システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、操作方法の異なる複数のデータベースのデータをオブジェクトとして同時に同じ操作方法で等価的に扱うためのデータベース統合アプリケーション構築システムに関する。

【0002】

【従来の技術】従来の情報システムでは、単一の巨大なデータベースに対してトランザクションベースで処理を行うのが主流であったが、計算機システムやネットワークの発達によって、複数のデータベースを同時に扱ったり、分散した複数のコンピュータ間でデータを検索したり更新したりすることが求められている。

【0003】従来のデータベースアプリケーションでは、図 1 2 のように、アプリケーション 1 0 0 がドライバ 1 0 1 を介して直接データベース 1 0 2 を操作するようにしている。また複雑化した企業内情報システムでは、各部署にさまざまな種類のデータベースが導入されているのが普通である。これらのデータベースの間には、通常、互換性がないため、複数のデータベースのデータを活用して処理を行う際には、全データを変換するための特別のプログラムを生成したり、互換性のある別のデータベースを導入し直したりしなければならない。

【0004】例えば、図 1 3 のように、データベース 1

4

0 2 A に加え、データベース 1 0 2 A とは互換性のないデータベース 1 0 2 B のデータをアプリケーション 1 0 0 で利用したい場合、データベース 1 0 2 B のためのドライバ 1 0 1 B と、ドライバ 1 0 1 B を操作する専用モジュール 1 0 3 を追加する。また図 1 4 のように、データベース 1 0 2 B のデータを既存のデータベース 1 0 2 A のデータ形式に変換するためのコンバートプログラム 1 0 4 を作る方法がある。

【0005】

【発明が解決しようとする課題】しかしながら、データベース毎に専用ドライバとモジュールを追加する図 1 3 のデータベース統合システムの構築にあつては、既存のデータベース用にアプリケーションが設計されているため、専用モジュールを設計するための工数が大きくなる傾向がある。

【0006】また図 1 4 のコンバートプログラムを使用するデータベース統合システムの構築は、コンバートプログラムでデータベース間のデータ変換を行う際に一部の情報が欠落することがある。またコンバートプログラムにより一旦変換したら、変換前のデータベースの機能を使ってデータを操作することはできないという不便さがある。

【0007】加えてデータベースが分散環境に対応していない場合、アプリケーションも分散環境には対応しない。更に図 1 4 のようなアーキテクチャに別のアプリケーションを統合しようとした場合は、手法が定まっていないためにアドホックな追加となつてしまい、保守性が悪い。一方、分散オブジェクト通信機構を実装したソフトウェアを使用することで複数のデータベースを統合することもできる。分散オブジェクト通信機構を実装したソフトウェアは、一般にオブジェクト・リクエスト・ブローカ (ORB: Object Request Broker) と呼ばれており、CORBA (オブジェクト指向技術の標準化団体米 OMG が固定した Common Object Request Broker Architecture) や分散 COM (米国マイクロソフト社の開発した Distributed Component Object Model) 等が知られている。

【0008】オブジェクト分散システムでは、あるプログラムが別のコンピュータにあるプログラムにネットワークを介して処理を依頼するものであり、このときオブジェクト・リクエスト・ブローカは、個々のオブジェクトがネットワーク上のどのコンピュータで動いているかを管理する。このようなオブジェクト・リクエスト・ブローカを使用して分散オブジェクトアーキテクチャをもつデータベース統合システムを構築した場合、アプリケーションの下位レベルにおいて、データベースが共通の通信プロトコルをサポートして通信を行うという共通化を行っている。

【0009】しかしながら、データベースのプロトコルも多様化し且つ複雑化しており、特定のプロトコルをサ

ポートするだけでは不十分である。そこで別のプロトコルをサポートしてしまうと、大掛かりな変更が必要となってしまう。更に、分散オブジェクト環境を実現するため、ある2つのコンピュータ間の通信の際に必ずオブジェクト・リクエスト・ブローカを通過しなければならない。このためブローカにリクエストが集中化し、完全な分散環境が構築しにくいという問題点がある。

【0010】本発明は、このような従来の問題点に鑑みてなされたもので、オブジェクトがどこのデータベースやサーバに格納されているか意識せずに等価的に扱うことができ、また分散アーキテクチャとして考えた場合、ブローカ等の特定部分にリクエストを集中化させずに完全な分散環境上で実現できるようにしたデータベース統合アプリケーション構築システムを提供することを目的とする。

【0011】

【課題を解決するための手段】図1は本発明の原理説明図である。まず本発明は、操作方法の異なる複数のデータベースに存在するデータをオブジェクトとして扱い、且つオブジェクト格納位置を意識せずに同じ操作方法で等価的に扱うためのデータベース統合アプリケーション構築システムを提供する。

【0012】即ち、本発明は、データベース、アプリケーション及び分散オブジェクトアーキテクチャといったミドルウェアを統一的に扱う仕組みを提供する。これにより、オブジェクトがどこのデータベースやサーバに格納されているか意識せずに等価的に扱うことができる。また、分散アーキテクチャとして考えた場合、ブローカ等でリクエストを集中化させず、完全な分散環境上で実現することができる。

【0013】本発明のデータベース統合アプリケーション構築システムは、基本アーキテクチャとして図1

(A)のように、論理層(Logical Layer)1、実現層(Implementational Layer)2、物理層(Physical Layer)3、及びプロトコル層(Protocol Layer)3を構築する。論理層1には、組織、文書等の複数種類の論理オブジェクト10A~10Bのロジックを実装してアプリケーションを構築する。

【0014】実現層2には、オブジェクトマネージャ12、共通オブジェクト14A~14D、及び論理IDをコネクタIDに変換する第1変換テーブル16を設ける。オブジェクトマネージャ12は、論理オブジェクト10A~10Bに対応して共通オブジェクト14A~14Aを生成すると共に論理オブジェクトの一意性を示す論理IDにより第1変換テーブル12を参照してコネクタIDを取得し、物理層3の提供するインタフェース24A~24Eを使用して共通オブジェクト14A~14Dにデータを操作させる。

【0015】物理層3には、複数のコネクタ18A~18Dを実装する。コネクタ18A~18Dは、実現層2

のコネクタIDで選択され、論理IDにより第2変換テーブル25を参照して得た物理IDによりデータベース側から実オブジェクトを取得してコネクタオブジェクト22A~22Dを生成し、コネクタオブジェクト22A~22Dを予め準備したインタフェース24A~24Eにより実現層2の共通オブジェクト14A~14Dを経由して論理層1の論理オブジェクト10A~10Dにリンクさせる。

【0016】即ち、コネクタ18A~18Dは、論理層1のアプリケーションが使用するデータベース32B、32C又はサーバ30A、30Dに格納されている実オブジェクトに対応し且つ実現層2の共通オブジェクトに1対1に対応している。またコネクタ18A~18Dは、固有のコネクタIDを有すると共にコネクタオブジェクト22A~22D、コネクタマネージャ20A~20D及び論理オブジェクトの論理IDを実オブジェクトの格納位置を示す物理IDに変換する第2変換テーブル25から構成される。

【0017】プロトコル層4は、コネクタ18A~18Dが操作するデータベース32B、32C又はサーバ30A、30Dとの通信を確立するアプリケーション・プログラム・インタフェース(API)を実装する。このような本発明のデータベース統合アプリケーション構築システムによれば、様々なデータベースやアプリケーションを、同じコンピュータであるローカル及び他のコンピュータであるリモートとも等価的に扱うことができるようになる。このため図1(B)のように、既存の資産を活用しながら共通のフレームワークのもとで業務アプリケーションなどを設計することが可能になり、開発期間を短縮することができる。また、使用しているデータベースやアプリケーションといったインフラストラクチャのアップグレードなどによる保守が容易になる。

【0018】物理層3のコネクタ18A~18Dは、処理要求を受け取ったときに起動する処理手続となる特定の機能をもったメソッドの集合でなる1又は複数のインタフェース24A~24Eを実装し、実装されているインタフェース24A~24Eを実現層2に公開する。物理層3のコネクタとしては、データベースコネクタ18B、18Cとサーバコネクタ18A、18Dの2種類が実装される。データベースコネクタ18B、18Cは、第2変換テーブル25から得られた物理IDで指定されるデータベース32B、32Cにアクセスして論理オブジェクト10B、10C論理IDで特定される実オブジェクトを取得する。

【0019】サーバコネクタ18A、18Dは、第2変換テーブル25から得られた物理IDで指定されるサーバ30A、30Dにアクセスして論理オブジェクト10A、10Dの論理IDで特定される実オブジェクトをサーバの管理するデータベースから取得する。またサーバコネクタ18A、18Dは、物理IDで特定されたサー

バ 3 0 A、3 0 D に論理 I D で指定される実オブジェクトを格納したデータベースが使用しているインタフェースを認識できないため、どのようなインタフェースかを問い合わせ、実現層 2 の共通オブジェクト 1 4 A、1 4 D に提示する。

【0 0 2 0】即ち、サーバコネクタ 1 8 A、1 8 D は第 2 変換テーブル 2 5 から取得した物理 I D により、リモートとなる他のコンピュータのオブジェクトを認識しているだけであり、論理 I D で指定されるリモートのデータベースの実オブジェクトがどのようなインタフェースをもっているかは、オブジェクトを格納しているデータベースのコンピュータでないと判らない。そこで、コネクタサーバ 1 8 A、1 8 D には、リモートのコンピュータにインタフェースを問い合わせる機能を設ける。

【0 0 2 1】コネクタ 1 8 A ~ 1 8 D が実装するインタフェース 2 4 A ~ 2 4 E は、基本となるデータへアクセスするためのインタフェース以外に、必要に応じて永続性機能、バージョン管理機能、ファイル操作機能、アクセス制御機能等のインタフェースを予め定義して実現層 2 の共通オブジェクト 1 4 A ~ 1 4 D により操作させる。

【0 0 2 2】データベース 3 2 B、3 2 C 又はサーバ 3 0 A、3 0 D のデータは、コネクタ 1 8 A ~ 1 8 D によりローカルのコンピュータのメモリ上にキャッシュされることでデータのアクセス時間を低減できる。コネクタ 1 8 A ~ 1 8 D は、データをコネクタオブジェクト 2 2 A ~ 2 2 D に変換する機能を有するため、オブジェクト指向に対応していないデータベースをオブジェクト指向データベースとして使用することを可能にする。

【0 0 2 3】実現層 2 のオブジェクトマネージャ 1 2 は、共通オブジェクト 1 4 A ~ 1 4 D を移動することにより、ローカルとなる同じコンピュータまたはリモートとなる異なるコンピュータのデータベース間のオブジェクトの移動を実現する。実現層 2 のオブジェクトマネージャ 1 2 は、共通オブジェクト 1 4 A ~ 1 4 D を複製することにより、同じコンピュータまたは異なるコンピュータのデータベース間のオブジェクトの複製を実現する。

【0 0 2 4】実現層 2 のオブジェクトマネージャ 1 2 は、物理層 3 のコネクタ 1 8 A ~ 1 8 D を動的に追加又は削除することにより、接続するデータベースやサーバを動作中に変更可能とする。コネクタ 1 8 A ~ 1 8 D に新機能を追加する場合、この新機能をサポートするインタフェースを設計してコネクタに実装する。コネクタ 1 8 A ~ 1 8 D で新機能としてサポートしたインタフェースを実現層 2 で操作できない場合、実現層 2 は新機能のインタフェースを論理層 1 に公開し、論理層 1 から実現層 2 を介さずに直接にコネクタ 1 8 A ~ 1 8 D のインタフェースを操作することを可能とする。

【0 0 2 5】論理層 1 の論理オブジェクトによって特定データを操作することにより、リモートとなる他のコン

ピュータのサービスを起動し、コンピュータ間で非同期にメッセージをやり取りしてサービス依頼側の待ち時間を低減させる。例えば実現層 1 にイベントオブジェクトを実装し、このイベントオブジェクトのプロパティ（資産）に所定の値を代入することにより、イベントが発生したことを他のコンピュータのイベントマネージャに伝えて対応するルールを起動させる。

【0 0 2 6】更に、論理層 1 に複数のオブジェクトを実装することにより、複数のデータベースやコンピュータにまたがって存在するデータを一度に検索・更新する所謂串刺しを実現する。

【0 0 2 7】

【発明の実施の形態】図 2 は本発明によるデータベース統合アプリケーション構築システムを実現するための基本アーキテクチャである。本発明にあっては基本アーキテクチャとして、論理層 1、実現層 2、物理層 3 及びプロトコル層 4 で構成される。論理層 1 はアプリケーション層として機能し、任意のアプリケーションのロジックを記述する。即ち、論理装置 1 に入や組織、文書といったビジネス上のオブジェクト 1 0 A、1 0 B、1 0 C、1 0 D 等を実装することで業務アプリケーションを構築することができる。

【0 0 2 8】実現層 2 には、オブジェクトマネージャ 1 2 と共通オブジェクト 1 4 A ~ 1 4 D という 2 種類のモジュールが存在する。オブジェクトマネージャ 1 2 は共通オブジェクト 1 4 A ~ 1 4 D の等価性を管理するため、

① 共通オブジェクトの生成；

② 論理 I D / コネクタ I D 変換テーブル（第 1 変換テーブル）1 6 の管理；を行う。

【0 0 2 9】共通オブジェクト 1 4 A ~ 1 4 D は、論理層 1 にアプリケーションロジックとして論理オブジェクト 1 0 A ~ 1 0 D が実装された際にオブジェクトマネージャ 1 2 により生成され、物理層 3 がインタフェースで提供するサービスを使って論理オブジェクト 1 0 A ~ 1 0 D のデータを操作する。また共通オブジェクト 1 4 A ~ 1 4 D が操作方法を知らない新たなインタフェースが物理層 3 に実装されたときのため、物理層 3 のインタフェースを論理層 1 に見せて、実現層 2 の介在することなく論理層 1 から直接、物理層 3 のインタフェースで提供するサービスを使ってデータを操作できる機構も備えている。

【0 0 3 0】物理層 3 は論理層 1 の論理オブジェクト 1 0 A ~ 1 0 D によるアプリケーションが対応しようとするデータベースやサーバをサポートするためのコネクタ 1 8 A ~ 1 8 D で示されたコネクタモジュールを実装する。コネクタ 1 8 A ~ 1 8 D は従来のデータベースやサーバへのドライバを拡張したモジュールであり、従来のドライバは設計のポリシーをもっていないが、本発明のコネクタ 1 8 A ~ 1 8 D には次の特徴がある。

【0 0 3 1】① 各コネクタ 1 8 A ~ 1 8 D は他のコネク

タと識別するためのコネクタ I D をもつ。

②コネクタ 1 8 A ~ 1 8 D はコネクタオブジェクト 2 2 A ~ 2 2 D とコネクタマネジャ 2 0 A ~ 2 0 D で構成される。

③コネクタオブジェクト 2 2 A ~ 2 2 D はデータベースやサーバ等に格納されている実オブジェクトに対応しており、且つ実現層 2 に生成された共通オブジェクト 1 4 A ~ 1 4 D に 1 対 1 に対応する。

【 0 0 3 2 】 ④コネクタマネジャ 2 0 は複数のコネクタオブジェクト 2 2 A ~ 2 2 D をまとめると共に、論理 I D / 物理 I D 変換テーブル ( 第 2 変換テーブル ) 2 5 の管理を行う。

⑤コネクタ 1 8 A ~ 1 8 D は 1 または複数のインタフェースをサポートする。

【 0 0 3 3 】 図 2 にあっては、コネクタオブジェクト 2 2 A がインタフェース 2 4 A をサポートし、コネクタオブジェクト 2 2 B がインタフェース 2 4 B、2 4 C をサポートし、コネクタオブジェクト 2 2 C がインタフェース 2 4 B、2 4 D、2 4 E をサポートし、更にコネクタオブジェクト 2 2 D がインタフェース 2 4 B、2 4 D、2 4 E をサポートしている。

【 0 0 3 4 】 インタフェース 2 4 A ~ 2 4 E は、ある特定の機能をもったメソッドの集まりであり、1 つのインタフェースがまとまった機能を提供する。ここでメソッドとは、オブジェクトが処理要求を受け取ったときに起動する処理手続きを意味する。コネクタオブジェクト 2 2 A ~ 2 2 D は自分がサポートできるインタフェース 2 4 A ~ 2 4 E を例えば図示のように実装し、これを実現層 2 に公開する。このため、コネクタオブジェクト 2 2 A ~ 2 2 D がある機能を実現できるかどうかインタフェース 2 4 A ~ 2 4 E を実装しているかどうかで実行時に判定することができる。

【 0 0 3 5 】 ここで、物理層 3 に設けたコネクタ 1 8 A ~ 1 8 D は、物理 I D で指定される実オブジェクトの格納先がサーバであればサーバコネクタと呼ばれ、データベースであればデータベースコネクタと呼ばれる。例えばコネクタ 1 8 A、1 8 D は実オブジェクトの格納先がサーバ 3 0 A、3 0 D であることから、サーバコネクタと呼ばれる。またコネクタ 1 8 B、1 8 C は実オブジェクトの格納先がデータベース 3 2 B、3 2 C であることから、データベースコネクタと呼ばれる。

【 0 0 3 6 】 物理層 3 に設けたコネクタ 1 8 A ~ 1 8 D が実装する最も基本となるインタフェースは、データへのアクセスを行うためのインタフェースである。このデータアクセス用のインタフェースは、原則として全てのコネクタ 1 8 A ~ 1 8 D がサポートする。このため、実現層 2 でオブジェクトマネジャ 1 2 が共通オブジェクト 1 4 A ~ 1 4 D を生成した際には、まずデータアクセス用のインタフェースをサポートしたコネクタをリンクとしてもつことになる。この基本インタフェースとしての

データアクセス用インタフェース以外に

①永続性機能

②バージョン管理機能

③ファイル操作機能

④アクセス制御機能

等をもつインタフェースを予めコネクタ 1 8 A ~ 1 8 D に対し必要に応じて定義することができ、これらのインタフェースが実現層 2 によって操作される。

【 0 0 3 7 】 更にコネクタ 1 8 A ~ 1 8 D に対しては、新しい機能をもつインタフェースを定義することも可能である。コネクタ 1 8 A ~ 1 8 D 側に新しい機能をもつインタフェースを定義した場合には、実現層 2 の共通オブジェクト 1 4 A ~ 1 4 D を実装し直さなければ、新しい機能をもつインタフェースを操作することはできない。

【 0 0 3 8 】 しかしながら、実現層 2 には物理層 3 におけるインタフェースを論理層 1 に見せる機能がある。このため、実現層 2 を新しい機能をもつインタフェースに対応して実装し直さず、論理層 1 から実現層 2 を介さずに直接、物理層 3 に定義された新たなインタフェースを使用することができる。プロトコル層 4 は物理層 3 のコネクタ 1 8 A ~ 1 8 D が操作するサーバ 3 0 A、3 0 D のアプリケーションやデータベース 3 2 B、3 2 C に特定化されたアプリケーション・プログラム・インタフェース ( A P I ) であり、従来のデータベース、サーバ用のドライバあるいは分散オブジェクト通信機構は、このプロトコル層 4 に位置する。

【 0 0 3 9 】 このような本発明の論理層 1、実現層 2、物理層 3 及びプロトコル層 4 からなる基本アーキテクチャでは、論理 I D、コネクタ I D 及び物理 I D という 3 種類の I D と、3 種類の I D 同士を関連付けるための論理 I D / コネクタ I D 変換テーブル 1 6 及び論理 I D / 物理 I D 変換テーブル 2 5 という 2 種類の変換テーブルを用いてオブジェクトの位置を特定する。ここで、3 種類の I D は次の内容をもつ。

【 0 0 4 0 】 まず論理 I D は、オブジェクトのグローバルな一意性を保証する I D であり、オブジェクトの格納場所や存在するコンピュータ等に依存しない。この論理 I D は実現層 2 のオブジェクトマネジャ 1 2 で発行され、オブジェクトマネジャ 1 2 が生成した共通オブジェクト 1 4 A ~ 1 4 D に付けられる。一度オブジェクトに付けられた論理 I D は、その後変更されることはない。

【 0 0 4 1 】 次にコネクタ I D は、あるコンピュータにおいて実オブジェクトが格納されているデータベースやサーバに対応したコネクタ 1 8 A ~ 1 8 D を示す I D である。具体的には、論理層 1 に実装した論理オブジェクト 1 0 B、1 0 C のように、その実オブジェクトがコンピュータ自身が管理しているデータベース 3 2 B、3 2 C に格納されている場合は、データベース 3 2 B、3 2 C のコネクタ I D となる。



【0042】また論理層の論理オブジェクト10A、10Dのように、その実オブジェクトが別のコンピュータとなるサーバ30A、30Dに格納されている場合は、別のコンピュータのコネクタIDとなる。この別のコンピュータのコネクタIDの場合は、必ずしもコネクタIDに対応したコンピュータにオブジェクトが格納されていなくともよく、更に別のコンピュータのコネクタIDで指定されるコンピュータにオブジェクトが格納されていてもよい。

【0043】ここで論理層1で生成した論理オブジェクト10A~10Dとその実オブジェクトが同じコンピュータに存在する場合は、このコンピュータをローカルマシンといい、これに対し別のコンピュータのデータベースやサーバに実オブジェクトが格納されている場合、その別のコンピュータをローカルマシンという。次に物理IDを説明する。物理IDはコネクタ18A~18Dの各々において、その実オブジェクトの格納位置を特定するためのIDであり、コネクタごとに物理IDの定義が違っていてもよい。例えば実オブジェクトがリレーショナルデータベースの場合、このリレーショナルデータベース内のデータを実オブジェクトするコネクタ内での物理IDは行列番号で示される。またディレクトリサーバをデータベースとして用いている場合、コネクタ内での物理IDはディレクトリパスで示される。

【0044】このような3種類の論理ID、コネクタID及び物理IDは、実現層2の論理ID/コネクタID変換テーブル16と物理層3の論理ID/物理ID変換テーブル25により相互に変換することができる。実現層2の論理ID/コネクタID変換テーブル16は、あるコンピュータ内で各オブジェクトがどのコネクタに存在するかを示しており、オブジェクトマネージャ12が管理している。また論理ID/物理ID変換テーブル25は、あるコネクタ内で各オブジェクトが実際にどこに格納されているかを示しており、各コネクタ18A~18Dのコネクタマネージャ20A~20Dで管理されている。

【0045】次にローカルマシンとなる他のコンピュータのオブジェクトを表すコネクタを説明する。この他のコンピュータとなるリモートマシンのデータベースに格納されたオブジェクトを表すコネクタは、サーバコネクタと呼ばれる。ローカルマシンとしての他のコンピュータに格納しているオブジェクトが実際にどんなインタフェースをもっているかは、ローカルマシンでないと分からない。このためサーバコネクタは、コネクタIDから求めた物理IDが示しているローカルマシンとしてのコンピュータにオブジェクトのもっているインタフェースを問い合わせる機能をもつ。

【0046】図3はローカルマシンにインタフェース問合せを行うためのサーバコネクタの説明図である。図3(A)は実現層2に共通オブジェクトが生成される前の

初期状態であり、物理層3にはサーバコネクタ18Aが実装されており、サーバコネクタ18Aはプロトコル層4のサーバAPI28Aを介して実オブジェクトの格納先を他のコンピュータとなるリモートマシンとしている。

【0047】この状態で図3(B)のように実現層2に共通オブジェクト14Aが生成され、実現層2における図2の論理ID/コネクタID変換テーブル16に対する論理IDの参照によるコネクタIDの取得により、物理層3のサーバコネクタ18Aが選択される。このサーバコネクタ18Aにおいて、コネクタマネージャ20Aが図2に示した論理ID/物理ID変換テーブル25により、オブジェクトの論理IDにより対応する物理IDを取得する。

【0048】この物理IDは、他のコンピュータとなるリモートマシン42のデータベースにオブジェクトが格納されることを示している。このためサーバコネクタ18Aは、プロトコル層4のサーバAPI28Aによりリモートマシン42のサーバAPI28Bにアクセスし、リモートマシン42の物理層3に存在するデータベースコネクタ18Bにコネクタオブジェクト22Bがサポートしているインタフェース24B、24C、24Dを問い合わせる。

【0049】この問い合わせによってローカルマシン40側は、リモートマシン42のオブジェクトを格納したデータベースがサポートする3種類のインタフェース24B、24C、24Dを認識し、図3(C)のようにコネクタオブジェクト22Aについて問合せた3つのインタフェース24B、24C、24Dを公開し、そのうちのインタフェース24Cを実現層2の共通オブジェクト14Aが操作してリモートマシン42側のデータベースのオブジェクトをアクセス可能とする。

【0050】図4は同一コンピュータ、即ちローカルマシンに存在する操作方法の異なる2つのデータベースをサポートする本発明のシステム構築の説明図である。図4(A)において、あるオブジェクト10Aが他のオブジェクト10Bをプロパティ参照データ34で示されるプロパティXで指し示しているとき、プロパティXで指し示されたオブジェクト10Bはデータベース32Bに格納されている時点では論理ID=B.1で示されている。

【0051】このため、論理層1にオブジェクト10Aがロードされ、プロパティ参照データ34のプロパティXを参照すると、図4(B)のようにプロパティXが指し示すオブジェクト10Bが自動的に生成され、そのデータがデータベース32Bから復元される。このための動作を説明すると次のようになる。図4(A)において、論理層1にオブジェクト10Aを実装すると、オブジェクトマネージャ12が実現層2に共通オブジェクト14Aを生成し、その論理ID=A.1による論理ID/コ

10

20

30

40

50

13

ネクタID変換テーブル16の参照によりコネクタID=A2を取得し、物理層3の対応するデータベースコネクタ18Aを選択する。

【0052】次に物理層3のコネクタオブジェクト22Aが提供するインタフェース24Aを使用し、プロトコル層4のデータベースドライバとしてのデータベースAPI28Aによりデータベース32Aから論理ID=A1で指定されるデータを実オブジェクトとして取得し、コネクタオブジェクト22Aを生成する。コネクタオブジェクト22Aは、共通オブジェクト14Aを介して論理オブジェクト10Aにリンクされ、データベース32Aのデータを操作できる。

【0053】このような論理層1の論理オブジェクト10Aの処理において、論理オブジェクト10Aのもつプロパティ参照データ34のプロパティXが他のオブジェクト10Bを差し示していたとする。実現層2の共通オブジェクト14AはプロパティXの値がオブジェクトの論理IDであることを認識した場合、論理ID/コネクタID変換テーブル16を参照して論理ID=B1に対応するコネクタID=B2を取得し、物理層3のデータ

ベースコネクタ18Bを選択する。

【0054】データベースコネクタ18Bのコネクタマネージャ20Bは、論理ID/物理ID変換テーブル25を参照して論理ID=B1に対応する物理ID=B3を取得し、これによってデータベースドライバ28Bを介してデータベース32Bから図4(B)のようにコネクタオブジェクト22Bを取得する。このようにデータベースコネクタ18Bにデータベース32Bよりコネクタオブジェクト22Bが読み込まれると、実現層2のオブジェクトマネージャ12はコネクタオブジェクト22Bに対応する実現層2の共通オブジェクト14Bと論理層1の論理オブジェクト10Bを生成し、物理層3に読み込んだコネクタオブジェクト22Bに関連付けるリンク張りを行う。

【0055】この結果、生成された実現層2の共通オブジェクト14Bに対しては、共通オブジェクト14AのプロパティXによる関連付けが行われ、オブジェクト10AのプロパティXは共通オブジェクトBを指すようになる。図5は図4のローカルに位置する2つのデータベース32A、32Bをサポートする本発明のシステム構築の処理動作のフローチャートである。まずステップS1で論理層1にオブジェクト10Aをロードし、オブジェクト10Aに基づくデータベース32Aに対する実現層2の共通オブジェクト14A及び物理層3のデータベースコネクタ18Aの選択によるコネクタオブジェクトAによるリンクが確立された状態で、ステップS2でオブジェクト10AによるプロパティXの参照が行われたとする。

【0056】このプロパティXの参照において、ステップS3でプロパティXが示す値Yを獲得し、オブジェ

14

クトマネージャ12はステップS4でプロパティ値Yはオブジェクトを表すか否かチェックする。具体的には、論理ID/コネクタID変換テーブル16を参照してコネクタIDが取得できなければ、プロパティ値Yはオブジェクトを表さないものと判断できる。

【0057】オブジェクトを表していない場合には、ステップS11でプロパティ値をそのまま論理オブジェクト10Aに返す。プロパティ値Yが論理ID/コネクタID変換テーブル16に登録されている場合には、オブジェクトを表すことから、ステップS4からステップS5に進み、オブジェクトを表すプロパティ値Yはメモリ上にキャッシュされたオブジェクトを表すか否かチェックする。

【0058】もしプロパティ値がメモリ上のオブジェクトを表している場合にはデータベース32Bからの取込みは必要とせず、メモリのキャッシングによりオブジェクト10Bを取得できることから、ステップS12でプロパティ値を論理オブジェクト10Aに返す。ステップS5でオブジェクトと判断されたプロパティ値Yがメモリ上にキャッシングされたオブジェクトでなかった場合には、ステップS6で実現層2のオブジェクトマネージャ12はプロパティ値Yで示される論理IDにより論理ID/コネクタID変換テーブル16を参照して、対応するコネクタIDを取得する。

【0059】続いて物理層3のデータベースコネクタ18BがコネクタIDにより選択され、ステップS7の物理層3のコネクタマネージャ20Bによるコネクタ内処理が実行される。ステップS7のコネクタ内処理は、図6にサブルーチンとしてのローカルコネクタ処理として示される。図6のローカルコネクタ処理にあつては、ステップS1で物理層3のコネクタマネージャ20Bが実現層2から提供された論理IDにより論理ID/物理ID変換テーブル25を参照し、実オブジェクトの格納先を示す物理IDを取得する。次にステップS2で、データベースドライバとしてのAPI28Bを介して物理IDの示す実オブジェクトをデータベース32Bから取得し、ステップS3で物理層3のコネクタマネージャ20Bが図4(B)のようにコネクタオブジェクト22Bを生成する。

【0060】続いてステップS4でコネクタオブジェクト22Bにデータベース32Bの物理オブジェクトをリンクし、ステップS5でコネクタオブジェクト22Bを実現層2のオブジェクトマネージャ12に返して一連のローカルコネクタ処理を終了する。このローカルコネクタ処理が終了すると、再び図5のメインルーチンにおけるステップS8に戻り、実現層2のオブジェクトマネージャ12が実現層2と論理層1にそれぞれ共通オブジェクト14Bと論理オブジェクト10Bを生成し、実現層の共通オブジェクト14Bを物理層3のコネクタオブジェクト22Bに関連付けるリンク付けを行う。最終的にステ

10

20

30

40

50

ップ S 1 0 で、解決された実現層 2 の共通オブジェクト 1 4 A から関連付けられ、プロパティ X の参照を通じて共通オブジェクト 1 4 B を共通オブジェクト 1 4 A に返すようになる。

【 0 0 6 1 】 図 7 は、あるオブジェクトが他のオブジェクトをあるプロパティで指しているとき、プロパティが指し示すオブジェクトが同一コンピュータとなるローカルマシンのデータベースではなく、別のコンピュータとなるリモートマシンのデータベースに格納されている場合の処理動作である。図 7 において、ローカルマシン 4 0 は、図 4 ( A ) の場合と同様にして、共通オブジェクト 1 4 A におけるプロパティ X の値が論理オブジェクト 1 0 B の論理 ID = B 1 を表すことで物理層 3 のサーバコネクタ 1 8 B - 1 を選択し、論理 ID / 物理 ID 変換テーブル 2 5 - 1 を論理 ID = B 1 で参照して得た物理 ID = B 3 がリモートマシン 4 2 を示している。

【 0 0 6 2 】 このためローカルマシン 4 0 のサーバコネクタ 1 8 C は、サーバ API 2 8 B - 1 によりリモートマシン 4 2 を物理 ID = B 3 で指定し、プロパティ X で示されるオブジェクト 1 0 B の論理 ID = B 1 を通知する。この論理 ID = B 1 を受けて、リモートマシン 4 2 の物理層 1 にあつては論理オブジェクト 1 0 B を生成し、これに伴い実現層 2 にオブジェクトマネージャ 1 2 が共通オブジェクト 1 4 B を生成し、論理 ID / コネクタ ID 変換テーブル 1 6 - 2 から論理 ID = B 1 に対応するコネクタ ID = B 2 0 を取得する。

【 0 0 6 3 】 このコネクタ ID = B 2 0 は物理層 3 に与えられ、データベースコネクタ 1 8 B - 2 が選択される。データベースコネクタ 1 8 B - 2 のコネクタマネージャ 2 0 B - 2 は、論理 ID / 物理 ID 変換テーブル 2 5 - 2 の参照により、論理 ID = B 1 から物理 ID = B 3 0 を取得し、これによってデータベースドライバ 2 8 B - 2 によりデータベース 3 2 B から物理 ID = B 3 0 に対応する物理オブジェクトを獲得し、コネクタオブジェクト 2 2 B - 2 を生成する。

【 0 0 6 4 】 このようにしてローカルマシン 4 0 の論理オブジェクト 1 0 A におけるプロパティ X の参照によるオブジェクト 1 0 B に対応するコネクタオブジェクト 2 2 B - 2 の生成が済むと、図 8 のようにローカルマシン 4 0 とリモートマシン 4 2 との間にリンク関係が確立され、図 4 ( B ) の場合と同様に、ローカルマシン 4 0 側にコネクタオブジェクト 2 2 B - 2 を読み込む。

【 0 0 6 5 】 更に実現層 2 及び論理層 1 に共通オブジェクト 1 4 B と論理オブジェクト 1 0 B - 1 を生成してコネクタオブジェクト 2 2 B 側と関連付け、最終的に共通オブジェクト 1 4 A のプロパティ X を共通オブジェクト 1 4 B に関連付けることで、ローカルマシン 4 0 の論理オブジェクト 1 0 A のプロパティ X によってリモートマシン 4 2 のデータベース 3 2 B のデータを論理オブジェクト 1 0 B によって操作することができる。

【 0 0 6 6 】 図 9 は、図 7 及び図 8 に示したリモートマシン 4 2 に対するローカルマシン 4 0 のリモート要求コネクタ処理のフローチャートであり、図 6 に示した同一コンピュータ装置内の処理となるローカルコネクタ処理と異なる点は、ステップ S 3 におけるテーブル参照で別コンピュータを示す物理 ID を取得し、ステップ S 2 で別コンピュータにオブジェクトの論理 ID を通知し、ステップ S 3 で論理 ID を示す物理オブジェクトを別コンピュータから取得して、ステップ S 4 でコネクタオブジェクトを生成する点であり、その他の点はローカルコネクタ処理と同じである。

【 0 0 6 7 】 図 1 0 は図 7 のリモートマシン 4 2 がローカルマシン 4 0 からオブジェクトの論理 ID を用いてオブジェクト要求を受けた場合のリモート応答コネクタ処理のフローチャートである。まずステップ S 1 で他のコンピュータからのオブジェクト要求に伴って通知されたオブジェクトの論理 ID に基づき、図 7 のように論理層 1 に例えば論理オブジェクト 1 0 B を生成する。

【 0 0 6 8 】 続いてステップ S 2 で実現層 2 のオブジェクトマネージャ 1 2 が論理 ID により変換テーブル 1 6 - 1 を参照してコネクタ ID を取得し、ステップ S 3 で物理層 3 のコネクタマネージャによるコネクタ内処理を実行する。このコネクタ内処理は同じコンピュータに接続されているデータベースから物理オブジェクトを取得する図 6 のローカルコネクタ処理と、更に他のコンピュータであるリモートマシンのデータベースから物理オブジェクトを取得する図 9 のリモート要求コネクタ処理のいずれかとなる。

【 0 0 6 9 】 ステップ S 3 でのコネクタ内処理により、コネクタ 1 8 B - 2 がデータベースから物理オブジェクトを取得してコネクタオブジェクト 2 2 B - 2 を生成すると、サーバ API 2 8 B - 3 を使用して要求元となるローカルマシン 4 0 のコネクタ 1 8 B - 1 にコネクタオブジェクト 2 2 B - 2 を転送してコネクタオブジェクト 2 2 B - 1 を生成させる。次にステップ S 5 でローカルマシン 4 0 のコネクタオブジェクト 2 2 B - 1 とリモートマシン 4 2 のコネクタオブジェクト 2 2 B - 2 の間にプロトコル層 4 のサーバ API 2 8 B - 1、2 8 B - 3 を経由した通信パスにより直接リンクを張る。最終的にステップ S 6 で不要となった論理層 1 の論理オブジェクト 1 0 B と実現層 1 2 の共通オブジェクト 1 4 B を削除する。

【 0 0 7 0 】 尚、図 7、図 8 にあつては、ローカルマシン 4 0 に対し 1 台のリモートマシン 4 2 を経由して実オブジェクトを取得する場合を例にとっているが、複数台のリモートマシン 4 2 を経由して最終的にデータベースの物理オブジェクトをローカルマシン 4 0 にコネクタオブジェクトとして生成することもできる。次に本発明のデータベース統合アプリケーション構築システムで実現される各種の機能を説明する。この機能は

- (1) オブジェクト指向データベースの実現機能
- (2) 分散オブジェクト環境の実現機能
- (3) 分散データベースの実現機能
- (4) キャッシュ処理機能
- (5) レプリケーション機能 (複製機能)
- (6) マイグレーション機能 (移動機能)
- (7) コネクタの動的追加機能
- (8) リモートサービス呼出し機能
- (9) コネクタ機能の追加機能
- (10) 串刺し操作機能

がある。

【0071】まずオブジェクト指向データベースの実現機能として、本発明にあっては、例えば図2において物理層3のコネクタ18A~18Dにおいて全てのデータはコネクタオブジェクト22A~22Dのようにオブジェクトで表される。このコネクタ18A~18Dの機能を利用すると、オブジェクト指向に対応していないデータベースのデータを、コネクタ18A~18Dでコネクタオブジェクト22A~22Dに変換することによりオブジェクト指向に対応させることができる。

【0072】次に分散オブジェクト環境の実現機能を説明する。図7、図8に示したように、ローカルマシン40が他のコンピュータとなるリモートマシン42を表すコネクタオブジェクト22Bをもつ場合のオブジェクトとのやり取りは、そのままりリモートマシン42との通信に反映されるため、分散オブジェクト環境を実現することとなる。

【0073】この場合、論理層1における論理オブジェクト10Bの操作は物理層3のコネクタオブジェクト22Bのもつインタフェース24Bの操作に分解されているため、特別にインタフェース定義言語 (IDL) を記述しなくとも分散化させることができる。次に分散データベースの実現機能を説明する。図7、図8のように他のコンピュータとなるリモートマシン42を表すローカルマシン40のコネクタオブジェクト22Bは、リモートマシン42のデータベースやアプリケーションのオブジェクトを表している。更にリモートマシン42自体が他のコンピュータをリモートマシンとして指定するコネクタオブジェクトをもつが、最終的にはデータベースやアプリケーションとなる。このリモートマシン42を表すコネクタオブジェクトを用いることで、分散環境に対応していないデータベースやアプリケーションを分散環境に対応させることができる。

【0074】次にキャッシュ処理機能を説明する。図2に示した物理層3のコネクタ18A~18Dにあっては、同一コンピュータとなるローカルマシン及び別コンピュータとなるリモートマシンのいずれについても、そのデータベースから取得したデータをローカルマシン上のメモリにキャッシュしておくことができる。メモリにキャッシュされたデータのアクセスは、実現層2のオブ

ジェクトマネージャ12がオブジェクトの論理IDによるメモリを参照してキャッシングされているか否か判定し、キャッシングされている場合には論理層1に論理IDを介してメモリ上のキャッシュアクセスを行わせる。

【0075】このキャッシュ処理機能によって、同一コンピュータとなるローカルマシンにあっては、磁気ディスク等の外部記憶装置からのオブジェクトの読み込みが不要となる。またリモートマシンのオブジェクトについてはコンピュータ間での通信を低減することができる。

次にレプリケーション (複製) 機能を説明する。図2の実現層2のオブジェクトマネージャ12は、共通オブジェクト14A~14Dの各々について同じ共通オブジェクトを複製するレプリケーションを実現することができる。この場合のレプリケーションによる共通オブジェクトの複製は、同一コンピュータとなるローカルマシン上のデータベース間でもよいし、リモートマシンとなる別コンピュータ間でもよい。

【0076】次にマイグレーション (移動) 機能を説明する。図2の実現層2のオブジェクトマネージャ12は、共通オブジェクト14A~14Dのマイグレーション

(移動) 機能を実現することができる。この共通オブジェクトの移動は、ローカルマシンの異なるデータベース間でもよいし、ローカルマシンとリモートマシンの異なるデータベース間でもよい。マイグレーションは基本的にはレプリケーションと同じであり、相違点はレプリケーションが複製後に複製前と複製後の両方のオブジェクトを残すのに対し、マイグレーションは複製前のオブジェクトを消去する点である。

【0077】次にコネクタの動的追加機能を説明する。実現層2のオブジェクトマネージャ12には共通オブジェクト14A~14Dを保守する機能があるため、物理層3に対しコネクタ18A~18Dを動的に追加したり削除したりすることができる。このコネクタの動的追加機能により、接続するデータベースやサーバを動作中に変更することができる。

【0078】次にリモートサービス呼出し機能を説明する。まず特殊なオブジェクトとしてイベントオブジェクトを準備する。イベントオブジェクトは、そのプロパティに値を代入するとイベントを発生したことがイベントマネージャに伝わり、対応するルールを起動することができる。例えば月末の日付をプロパティの値として代入すると、時間管理処理により設定日時に達したときに月報作成ルールを起動する。

【0079】この場合にローカルマシンに存在するイベントオブジェクトにプロパティの値を代入することで、リモートマシンのサービスの呼出しを行うことができる。このリモートサービスの呼出し操作はリモートマシンからの戻り値を必ずしも要求しないため、コンピュータ間の通信を非同期通信にでき、リモートサービスを依頼したローカルマシンにおける待ち時間を最小にするこ

とができる。

【 0 0 8 0 】次にコネクタ追加機能を説明する。図 2 の物理層 3 において、既存のコネクタ 1 8 A ~ 1 8 D に対し新しい機能をもつコネクタを追加する場合、既存のコネクタをソースレベルで拡張して新しい機能を実装して新しいコネクタとするか、既存のコネクタをソースレベルで変更せずに使用してその既存のコネクタの機能を公開し、且つ新しい機能を実装したコネクタを実装して新しいコネクタとする。

【 0 0 8 1 】コネクタ新機能の追加に伴う新たなインタフェースを実装した場合、実現層 2 の共通オブジェクトの修正を必要とするが、共通オブジェクトを修正せずに新たにサポートしたインタフェースを論理層 1 に見せることで、実現層 2 を介さずに直接、論理層 1 の論理オブジェクトから物理層 3 のインタフェースを操作することができる。

【 0 0 8 2 】次に串刺し操作機能を説明する。複数のデータベースやリモートマシンとしてのコンピュータにまたがるオブジェクトに対する操作は、同一種類のオブジェクトに対する同一操作を複数回行う操作として実装することができる。この場合、アプリケーションのロジックはローカルマシンとなる論理層 1 に実装すればよい。これにより、複数のデータベースやリモートマシンとしてのコンピュータにまたがって存在するデータを一度に検索したり更新したりすることができる。

【 0 0 8 3 】図 1 1 は本発明をインフラストラクチャとして業務アプリケーションを構築した場合の具体的な実施形態の説明図である。図 1 1 において、サーバ 5 0 の論理層 1 にはユーザ「松塚」のオブジェクト 1 0 A がオブジェクト 1 0 B としてのワークフローの作業内容に従ってオブジェクト 1 0 C としての文書を編集している環境である。ユーザを表すオブジェクト 1 0 A は、データベース 3 0 A のディレクトリサーバ、サーバ 3 2 B のワープロソフトウェア、更にサーバ 5 2 側に存在するデータベース 3 0 D の文書データベースのオブジェクトを、物理層 3 にコネクタオブジェクト 2 2 A、2 2 B 及び 2 2 C として読み込み、実現層 2 の共通オブジェクト 1 4 A、1 4 B、1 4 C の連携により使いながら処理を行っている。

【 0 0 8 4 】

【発明の効果】以上説明してきたように本発明によれば、様々なデータベース、サーバ、更にはアプリケーションをローカル及びリモートのいずれについても等価的に扱うことができ、この結果、既存の資産を活用しながら共通のフレームワークのもとで業務アプリケーション等を設計することが可能となり、業務アプリケーションを実現する情報システムの開発期間を大幅に短縮できる。

【 0 0 8 5 】また使用しているデータベースやアプリケーションといったインフラストラクチャのアップグレー

ド等による保守が容易にできる利点を有する。

【図面の簡単な説明】

【図 1】本発明の原理説明図

【図 2】本発明の基本アーキテクチャの説明図

【図 3】本発明で用いるサーバコネクタによるリモートへのインタフェース問合せ機能の説明図

【図 4】ローカルに位置する 2 つのデータベースをサポートする本発明のシステム構築の説明図

【図 5】図 4 のシステム構築の処理動作のフローチャート

【図 6】図 4 のローカルのデータベースに対するコネクタ処理のフローチャート

【図 7】リモートのデータベースをサポートする本発明によるシステム構築の説明図

【図 8】図 7 でシステムを構築した後のリモートのデータベースの参照動作の説明図

【図 9】図 7 のリモートのデータベースに対するローカルでのコネクタ処理のフローチャート

【図 1 0】ローカルからの処理要求に対するリモートでのコネクタ処理のフローチャート

【図 1 1】本発明をインフラストラクチャとして業務アプリケーションを構築した場合の説明図

【図 1 2】従来の基本的なデータベースをサポートするシステムの説明図

【図 1 3】専用モジュールにより互換性のない 2 つのデータベースをサポートする従来システムの説明図

【図 1 4】コンバートプログラムにより互換性のない 2 つのデータベースをサポートする従来システムの説明図

【符号の説明】

1 : 論理層 (アプリケーション層)

2 : 実現層

3 : 物理層

4 : プロトコル層

1 0 A ~ 1 0 D : 論理オブジェクト (アプリケーション)

1 2 : オブジェクトマネージャ

1 4 A ~ 1 4 D : 共通オブジェクト

1 6 : 論理 ID / コネクタ ID 変換テーブル (第 1 変換テーブル)

1 8 A ~ 1 8 D : コネクタモジュール (コネクタ)

2 0 A ~ 2 0 D : コネクタマネージャ

2 2 A ~ 2 2 D : コネクタオブジェクト

2 4 A ~ 2 4 E : インタフェース

2 5 : 論理 ID / 物理 ID 変換テーブル (第 2 変換テーブル)

2 8 A ~ 2 8 D : アプリケーション・プログラム・インタフェース (API)

3 0 A、3 0 D : サーバ

3 2 B、3 2 C : データベース

3 4 : プロパティ参照データ

10

20

30

40

50

21

22

36 : プロパティ値

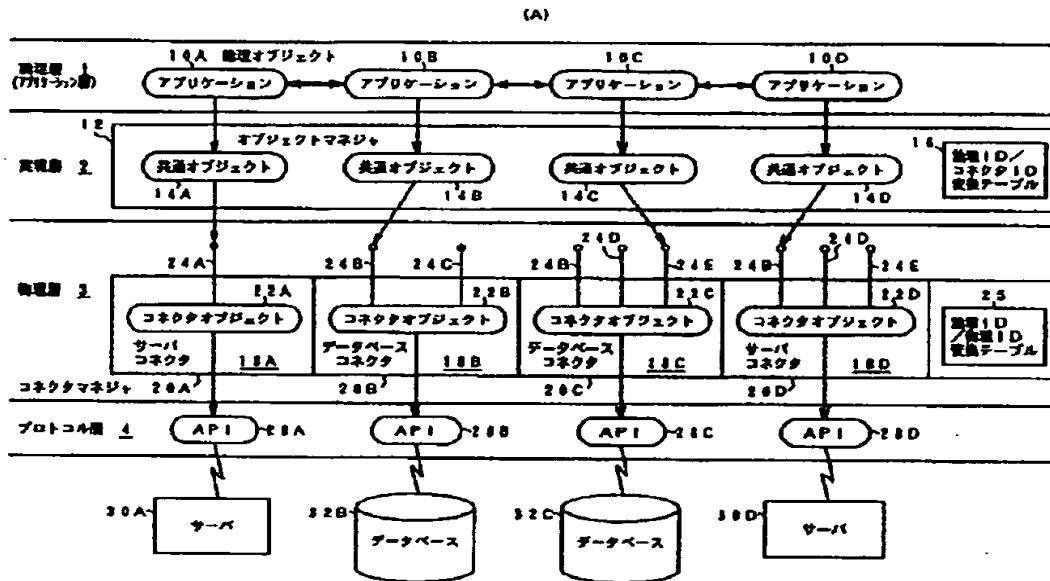
42 : リモートマシン (リモートコンピュータ)

40 : ローカルマシン (ローカルコンピュータ)

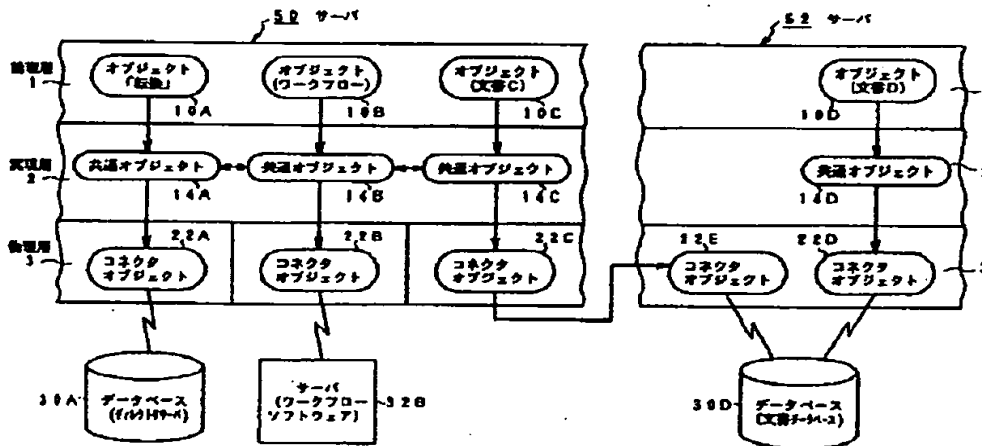
50, 52 : サーバ

【 図 1 】

本発明の原理説明図



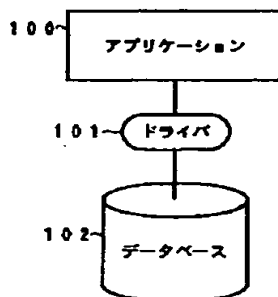
(B)



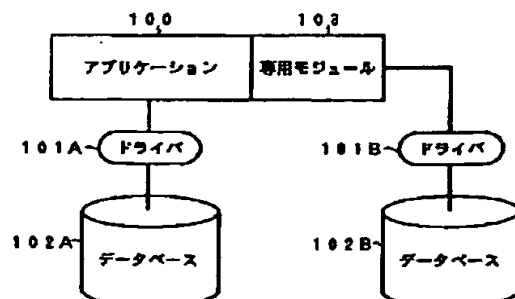
【 図 12 】

【 図 13 】

従来の基本的なデータベースをサポートするシステムの説明図

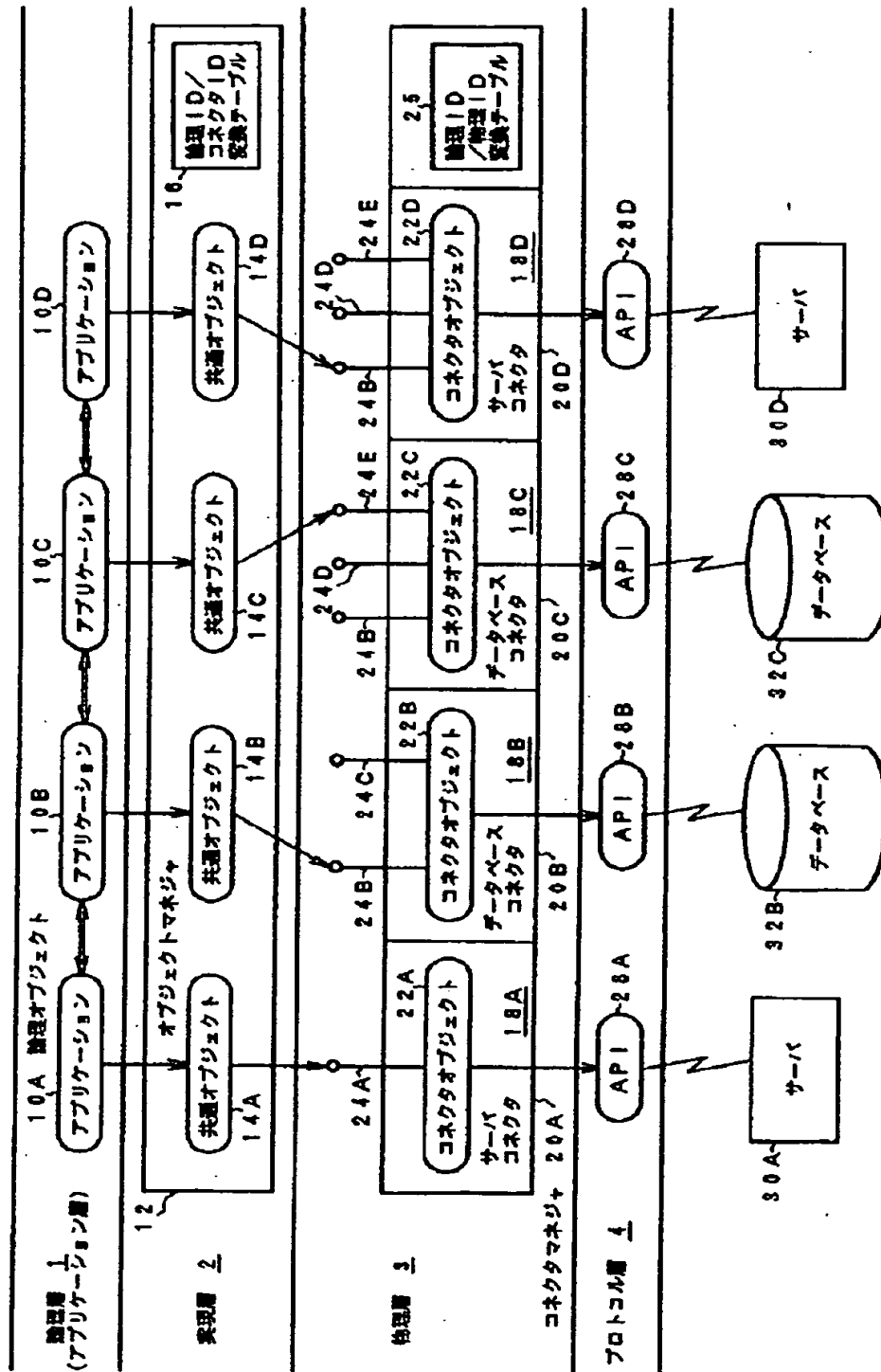


専用モジュールにより互換性のない2つのデータベースをサポートする従来のシステムの説明図



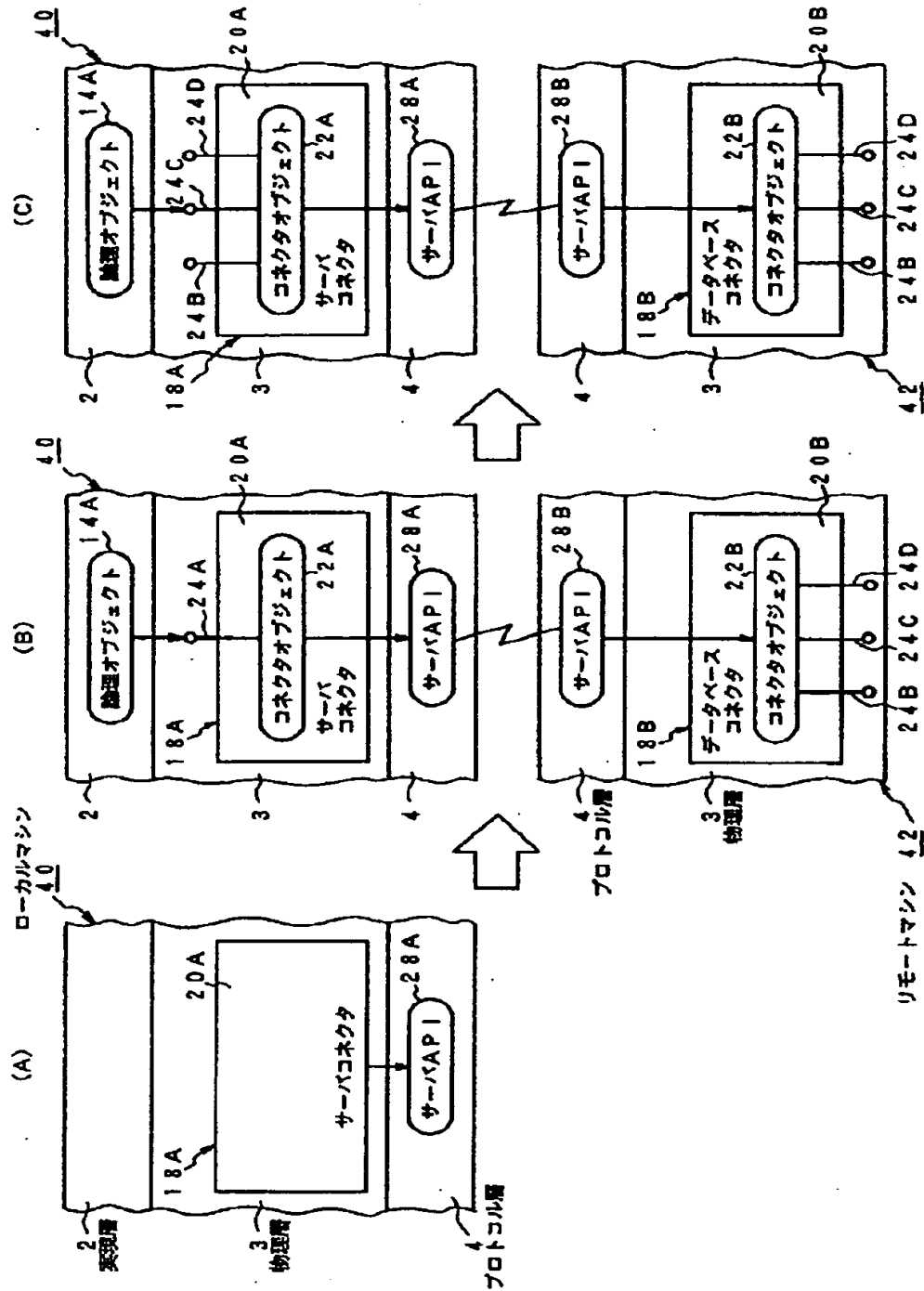
【 図 2 】

本発明の基本アーキテクチャの説明図



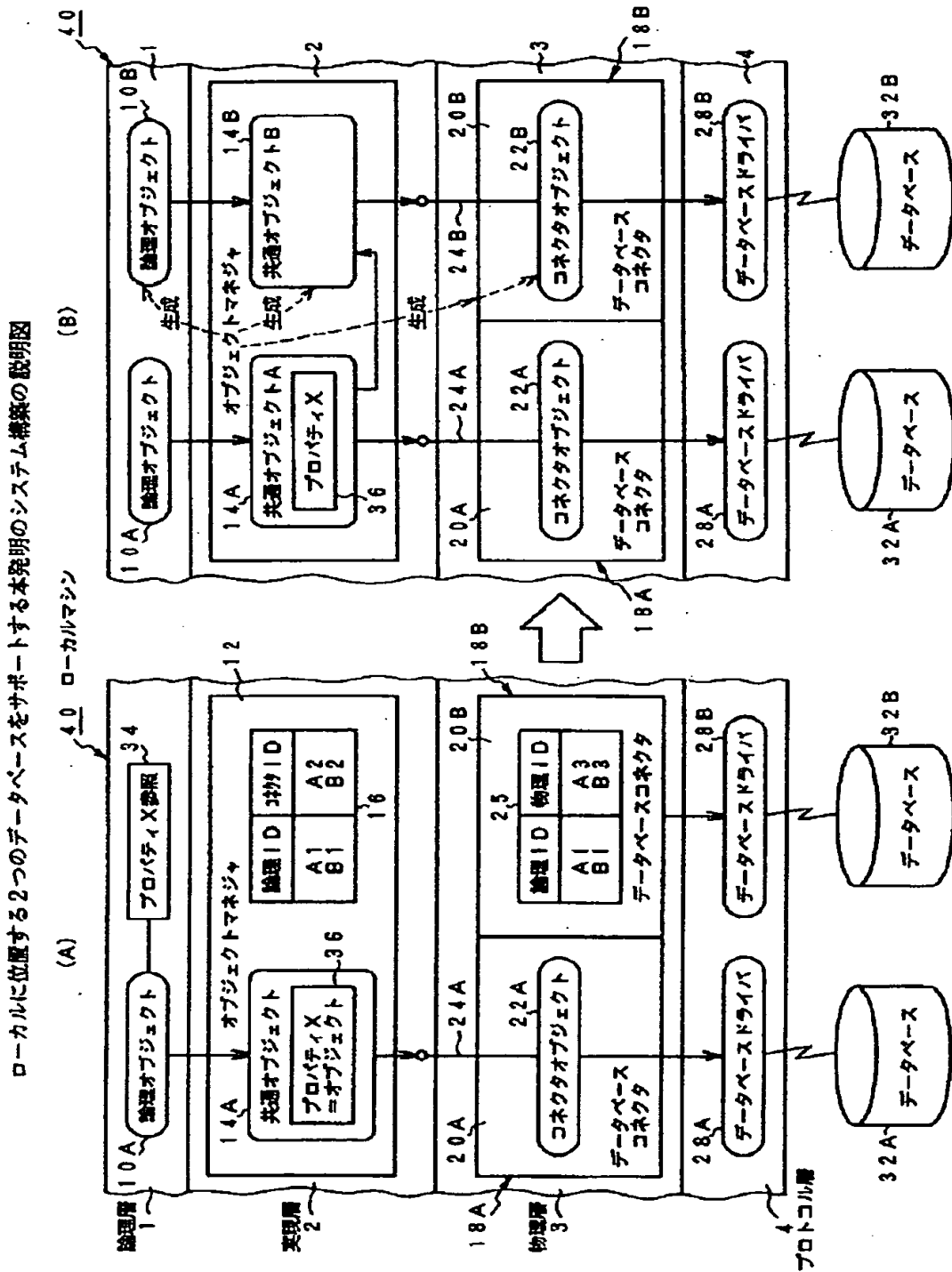
【 図 3 】

本発明で用いるサーバコネクタによるリモートへのインタフェース問合せ機能の説明図



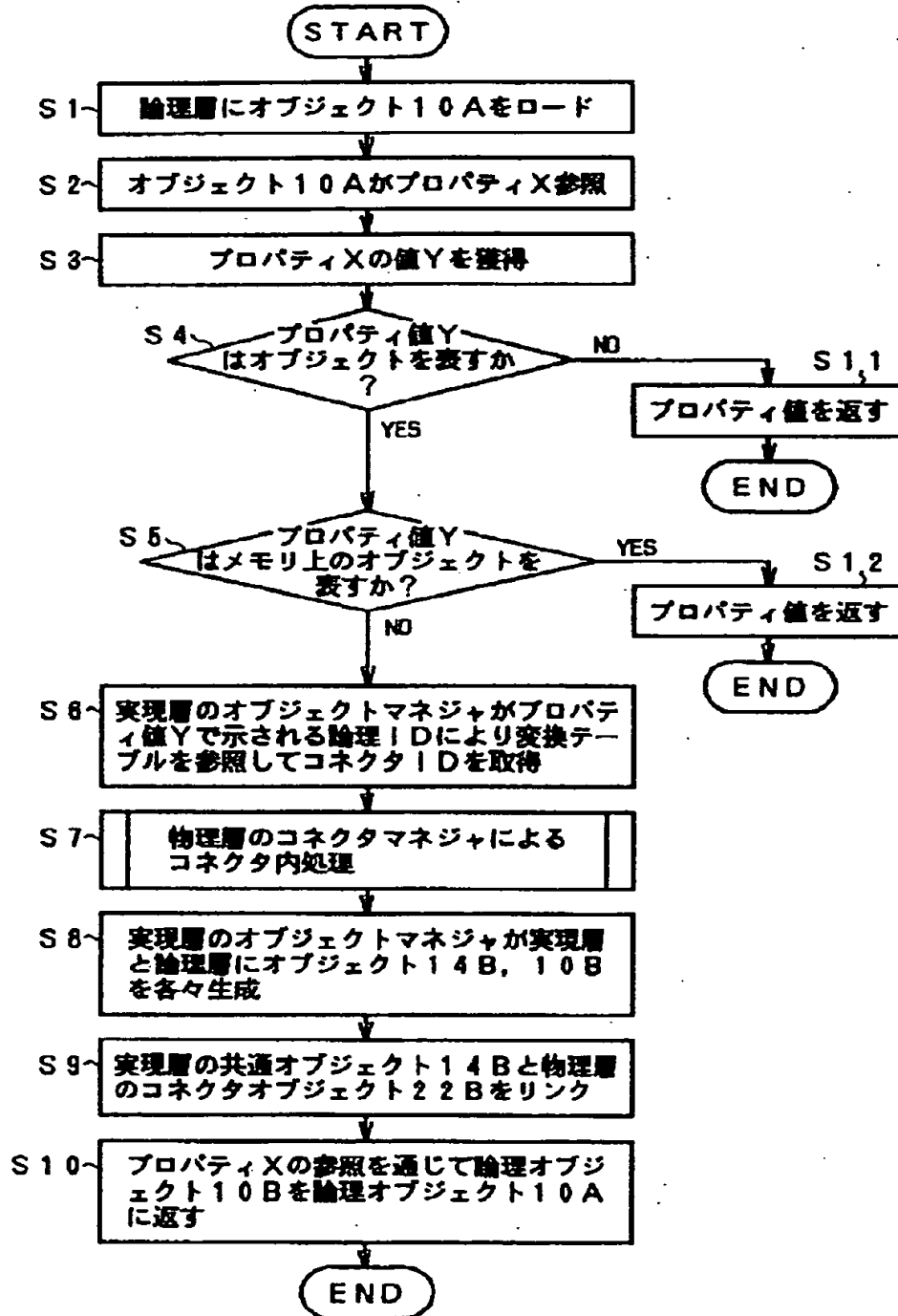


【 図 4 】



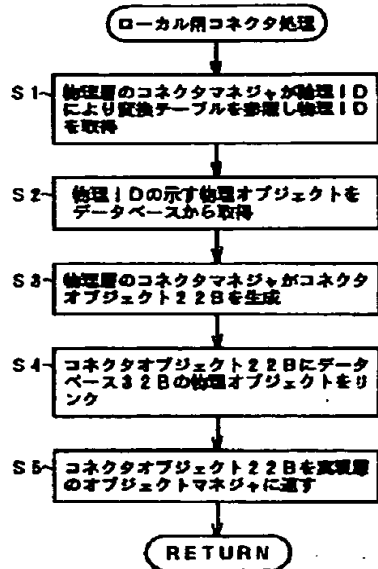
【図 5】

図 4 のシステム構築の処理動作のフローチャート



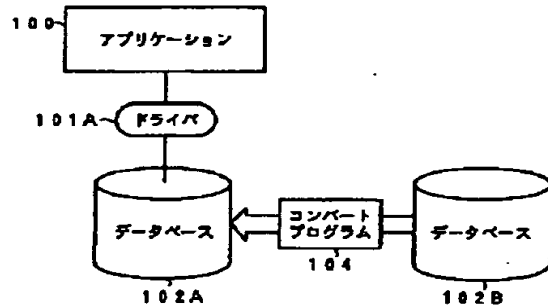
【 図 6 】

図4のローカルのデータベースに対するコネクタ処理のフローチャート



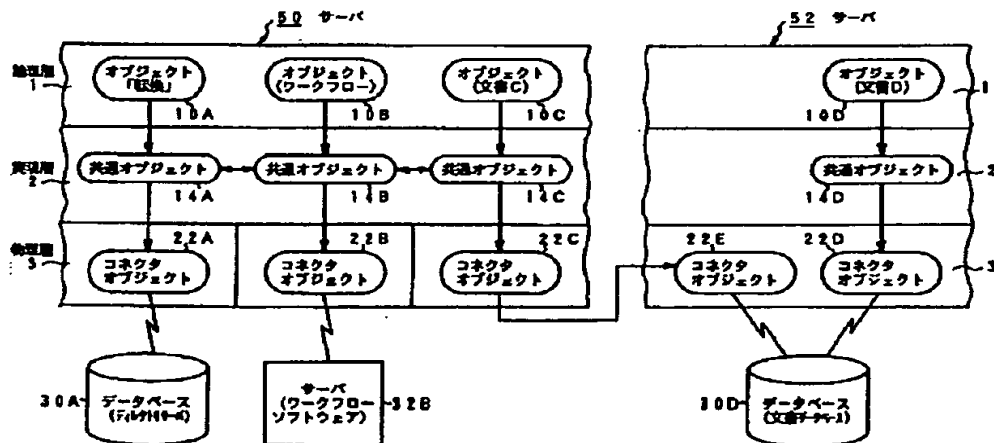
【 図 1 4 】

コンバートプログラムにより互換性のない2つのデータベースをサポートする従来システムの説明図

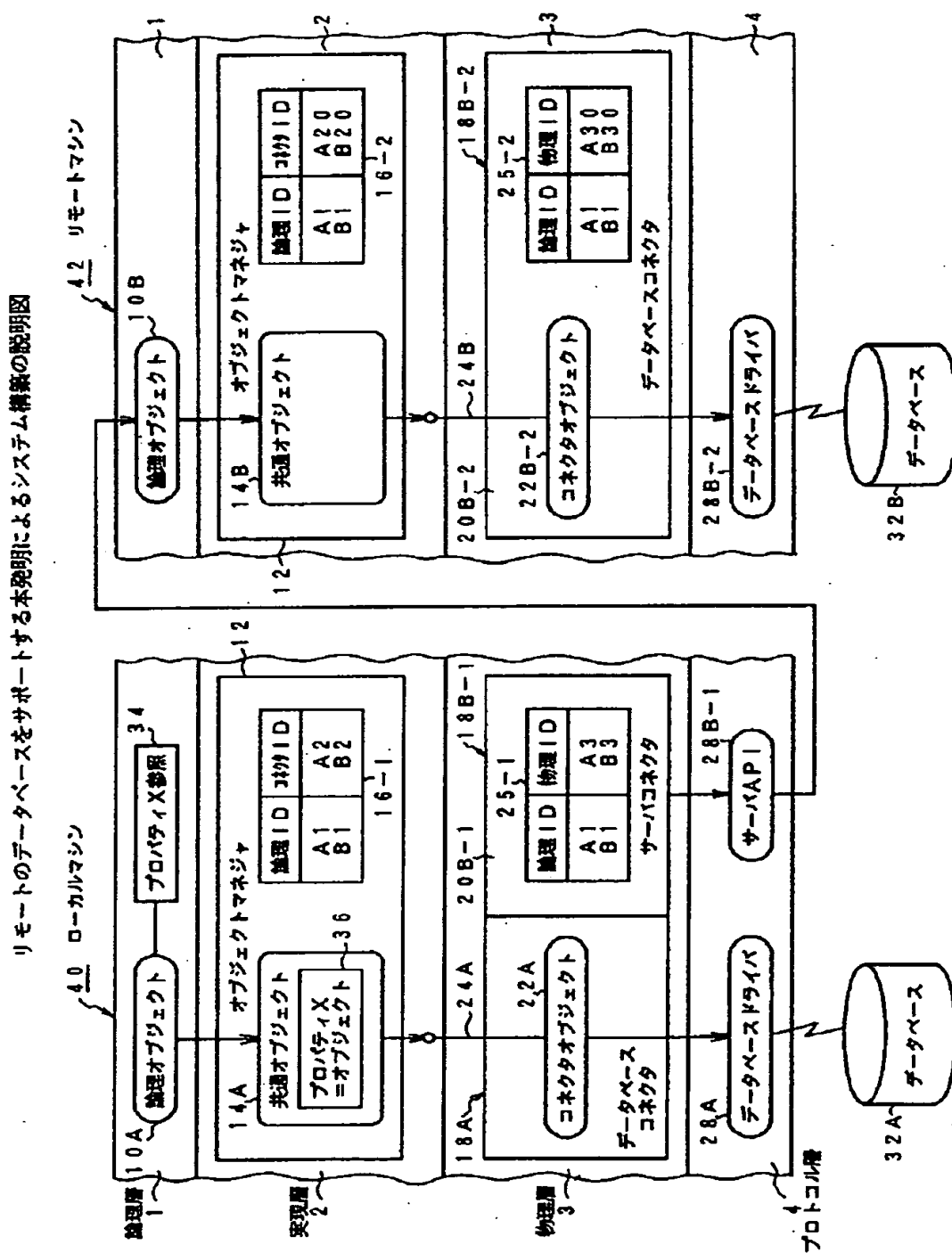


【 図 1 1 】

本発明をインフラストラクチャとして業務アプリケーションを構築した場合の説明図



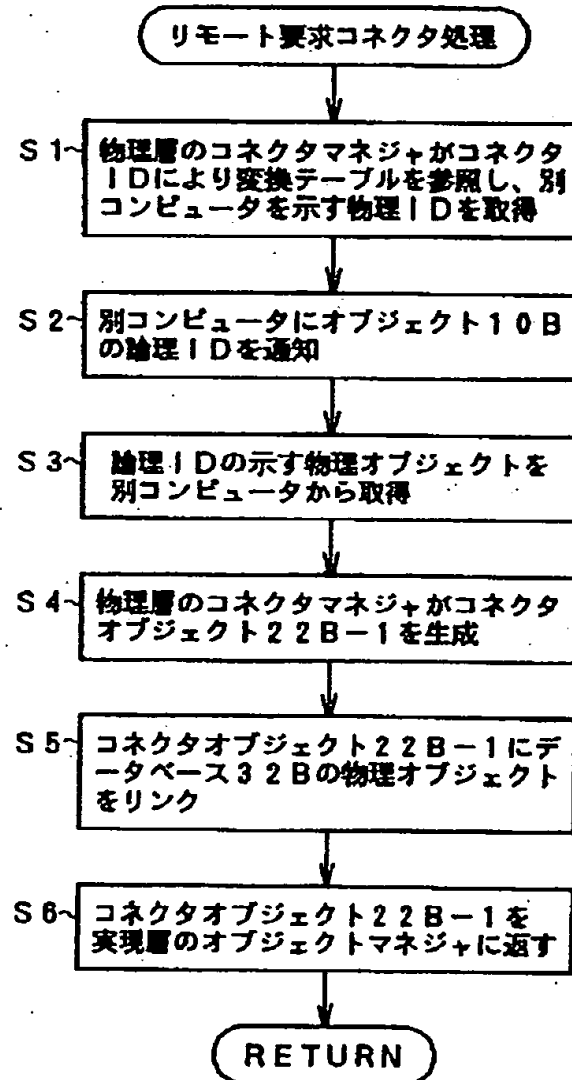
【 図 7 】





【図9】

図7のリモートのデータベースに対するローカルでのコネクタ処理のフローチャート



【図 1 0】

ローカルからの処理要求に対するリモートでのコネクタ処理のフローチャート

